



Análise da influência de alterações em parâmetros do *branch and cut* em um problema de designação generalizada

Arthur Medeiros Figueiredo Barreto
UNESP Bauru
arthur.medeiros@unesp.br

Matheus Artioli Leandrin
UNESP Bauru
matheus_leandrin@yahoo.com.br

Edilaine Martins Soler
UNESP Bauru
edilaine@fc.unesp.br

Adriana Cristina Cherri Nicola
UNESP Bauru
adriana@fc.unesp.br

Resumo: O problema de designação generalizada por definição consiste em atribuir tarefas à agentes. Por se tratar de um problema com inúmeras aplicações práticas, neste trabalho o problema será utilizado para a designação generalizada de múltiplos itens para diferentes armazéns utilizando uma situação real. O objetivo é verificar a influência dos parâmetros do método *branch and cut* na seleção de nós e na seleção dos planos de corte, além da influência no tempo computacional e otimalidade do problema, através de métodos exatos de Programação Inteira Mista. Com os testes computacionais realizados, foi possível verificar que utilizar *solvers* automaticamente, sem alterar as configurações padrões, pode ser um bom procedimento quando não se tem conhecimento sobre as especificações do problema.

Palavras-chave: Branch and Cut. Planos de Corte. Designação Generalizada. Otimização.

1. Introdução

A pesquisa operacional é uma ciência exata que estuda a otimização de problemas cotidianos através de modelagens matemáticas e diferentes métodos de solução. A utilização da pesquisa operacional auxilia no processo de tomada de decisão através da observação, coleta de dados e resolução de problemas, obtendo soluções ótimas ou boas o suficiente, através de métodos exatos ou heurísticos, respectivamente. O problema de designação é um exemplo de aplicação da pesquisa operacional amplamente discutido na literatura e objeto de estudo deste trabalho.

O problema de designação compreende a designação de tarefas a agentes, o qual pode ser visualizado em diversas situações, como em máquinas a serem destinadas a determinadas localidades e funcionários a serem alocados a tarefas. Arenales et al. (2006) e Hillier e Lieberman (2012) classificam o problema de designação como um caso especial de programação linear (problema clássico).

No entanto, uma das características desses problemas clássicos é a complexidade. A complexidade é um fator que está associado às restrições dos problemas e principalmente ao número de variáveis geradas no processo de resolução. O problema de designação é um problema combinatório e complexo quando o tempo exigido para a resolução computacional torna-se uma restrição do problema. A partir disso os pesquisadores buscam propor métodos matemáticos capazes de indicar soluções adequadas em tempos aceitáveis, como por exemplo o *branch and cut* (B&C).

Qin et al. (2014) definem o método *branch and cut* como uma derivação ou extensão do método *branch and bound* (B&B) elaborado em Land e Doig (1960). No B&B o problema é relaxado, ou seja, descarta-se todas as condições de integralidade das variáveis e, em seguida, o problema é resolvido de forma contínua, obtendo-se valores fracionários para as variáveis com a relaxação. O próximo passo consiste em diminuir a região factível através da limitação dos valores obtidos pelas variáveis fracionárias. O procedimento é repetido e assim são gerados inúmeros subproblemas, chamados de nós no método B&B, formando uma estrutura semelhante a uma árvore. O algoritmo atinge seu fim quando a solução ótima do subproblema é inteira, quando o subproblema é infactível ou quando são encontradas apenas soluções piores, ocasionando a parada por otimalidade, infactibilidade ou por qualidade respectivamente.

O método *branch and cut* foi introduzido em 1987 como resolução inicial para o problema do caixeiro viajante. (Sousa e Asada, 2012). Conforme descrito em Qin et al. (2014), no B&C é mantida a estrutura da árvore de nós do B&B, porém desta vez com o acréscimo de planos de corte, chamados de desigualdades, buscando diminuir a quantidade de nós a serem explorados e, consequentemente, atingir um limite mais restrito. Neste trabalho foram utilizados 6 tipos de planos de corte: Clique, Cobertura (*Cover cuts*), Cobertura de Fluxo (*Flow cuts*), Arredondamento de Número Inteiro Combinado (MIR), Metade Zero (*Zero half cuts*) e Cortes fracionários de Gomory.

Tanto no B&C quanto no B&B deve-se escolher qual subproblema, ou seja, qual nó será analisado a medida que se percorre a árvore de soluções. Existem inúmeras técnicas na literatura para escolha do nó. Abordamos dois dos procedimentos clássicos mais utilizados: busca pelo melhor limitante e a busca por profundidade (El Afia e Kabbaj, 2017). Na busca pelo melhor limitante se dá prioridade ao subproblema com melhor função objetivo e na busca por profundidade os últimos nós incluídos na árvore são analisados. Outras técnicas de escolhas de nós também podem ser aplicadas, como em He et al. (2014).

Em suma, este trabalho busca estudar a influência dos planos de corte e do critério de escolha de nós do B&C na resolução de um problema de designação generalizada, proposto inicialmente em Barreto, Araújo e Pisco (2016). São realizadas combinações na escolha dos métodos para verificar quais deles têm mais influência sobre o modelo em questão. Como forma de interpretação dos resultados, diversos parâmetros finais são mensurados, como o tempo computacional, o número de soluções encontradas e o número de iterações.

A divisão deste trabalho é dada a seguir. A Seção 2 apresenta o modelo matemático para o problema de designação generalizada. Na Seção 3 apresentamos os métodos utilizados para resolução do problema. Na Seção 4 são apresentados os resultados e discussões, seguidos pela conclusão, na Seção 5.

2. Modelo matemático

Segundo Arenales et al. (2006) a designação pode ser comum, generalizada, generalizada com múltiplos níveis ou generalizada com múltiplos recursos. O que diferencia a designação comum da generalizada é a dimensão do problema. Na forma mais simples, tem-se n tarefas a serem alocadas para n agentes. Na forma generalizada, o problema consiste em alocar m agentes para n tarefas.

O problema abordado neste trabalho e inicialmente descrito em Barreto, Araújo e Pisco (2016) trata-se de uma situação real, em que aborda-se a necessidade de realocação de itens para outros locais, devido o término das atividades em uma loja. É necessário designar estes itens para cada um dos armazéns disponíveis, minimizando a distância e respeitando a capacidade disponível em cada um dos locais de destino.

$$\text{Min} \sum_{i=1}^m \sum_{j=1}^n C_{ij} X_{ij} \quad (1)$$

$$\sum_{i=1}^n X_{ij} = 1 \quad \forall j \quad (2)$$

$$\sum_{i=1}^n A_{ij}X_{ij} \leq b_i \quad \forall i \quad (3)$$

$$X_{ij} \in 0, 1 \quad \forall j, \forall i \quad (4)$$

No modelo (1)-(4), a equação (1) minimiza o custo total de atribuição dos agentes às tarefas que, neste caso, corresponde a minimizar as distâncias dos armazéns para a loja. O conjunto de restrições (2) faz com que cada tarefa seja executada por um único agente. As restrições (3) referem-se a capacidade, em que b_i é a capacidade de cada agente i e esta não deve ser excedida pelo fator de cada tarefa A_{ij} . As restrições (4) garantem o domínio das variáveis.

3. Métodos

Foi utilizado o método exato de resolução para esta aplicação do problema de designação através do *solver* CPLEX. Uma instância de 45 armazéns com 200 itens foi utilizada. No entanto, para evitar longos tempos de processamento, um GAP de 5% foi introduzido no programa, ou seja, ao se atingir uma solução factível com pelo menos 5% de diferença em relação ao melhor limite inferior conhecido, a execução é interrompida.

O problema foi solucionado através do método *branch and cut*. Com relação a escolha de nós, foram considerados dois parâmetros: busca por profundidade e melhor limitante.

O processo de resolução foi dividido em duas etapas. A primeira consiste em resolver o problema de designação generalizada através do B&C com o CPLEX em modo automático. Desta forma, foi possível obter os planos de corte utilizados pelo programa para, na segunda etapa, utilizar novamente o B&C com cada um destes planos individualmente. Para cada teste de um plano de corte, os dois critérios de escolha de nós foram aplicados separadamente, permitindo assim uma comparação entre o critério de escolha de nós.

4. Resultados e discussões

O programa foi executado em um computador com processador i5-5200U, 8 GB RAM DDR3L. Para execução do código foram utilizados os *softwares* Cplex 12.7 da IBM e o Visual Studio 2013 da Microsoft.

Os resultados obtidos estão divididos em 3 grupos. O primeiro grupo corresponde a resolução do problema de forma automática, em que o CPLEX escolhe quais planos de corte e quantas vezes eles devem ser usados. Os grupos 2 e 3 apresentam cada um dos planos de corte ao se considerar como critério na escolha de nós a busca pelo melhor limitante e profundidade, respectivamente. Os resultados para o grupo 1 estão apresentados nas Tabelas 1 e 2. A Tabela 1 apresenta os resultados para os parâmetros analisados. A Tabela 2 apresenta os tipos de corte escolhidos pelo CPLEX e o número de desigualdades utilizadas.

Tabela 1: Parâmetros encontrados na resolução automática

	Tempo (s)	Solução encontrada	Número de soluções encontradas	Nós explorados	Nós restantes	Iterações
Limitante	276,89	774	32	816668	66737	2130261
Profundidade	272,34	774	58	261994	26313	2009427

Tabela 2: Número de desigualdades geradas

	Clique	Cobertura	Cobertura de fluxo	MIR	Zero	Gomory
Limitante	6	566	20	80	40	7
Profundidade	4	299	17	74	39	6

Embora os dois métodos tenham atingido o mesmo resultado na função objetivo, 774, o número de possíveis soluções encontradas é maior para a busca em profundidade, isto porque quanto mais se aprofunda na árvore de soluções, maior a chance de se encontrar uma solução para o problema, mesmo que não seja a ótima. A diferença em tempo computacional entre os dois métodos foi inferior a 2%. Cortes por cobertura e MIR foram responsáveis pelo maior número de desigualdades, concluindo-se que, quando utilizados separadamente, são os mais benéficos no tempo de resolução. Para verificar-se os benefícios no tempo de resolução, todos os cortes foram testados separadamente e estão representados nos grupos 2 e 3. O grupo 2, apresentado na Tabela 3, refere-se a busca pelo melhor limitante utilizando cada um dos cortes separadamente.

Tabela 3: Resultados do Grupo 2 - busca pelo melhor limitante

Planos de corte	Tempo (s)	Solução encontrada	Número de soluções encontradas	Nós explorados	Nós restantes	Iterações	Número de desigualdades
Clique	23469,49	765	109	9197678	8551454	44185748	12
Cobertura	190,83	775	29	74144	53726	1555595	711
Cobertura de fluxo	1152,61	762	104	601379	319286	5705882	128
MIR	221	774	23	51419	43455	1476101	738
Metade zero	157,58	770	21	63745	41802	1190929	471
Fracionários de Gomory	17622,21	758	125	7387104	6512160	40102190	50

Os dois maiores tempos de resolução do grupo 2 pertencem aos cortes que geraram menos desigualdades quando o CPLEX resolveu automaticamente: clique e cortes de Gomory, conforme a Tabela 2. Isto pode indicar que estes cortes não são tão relevantes para este problema específico e, forçar a sua utilização, não apresenta benefícios computacionais para o problema.

Os 3 maiores tempos de resolução (Clique, Flow e Cortes de Gomory) também apresentaram os maiores números de soluções encontradas. Isto já era esperado pois quanto maior o tempo, maior a quantidade de nós explorados e, conseqüentemente, maior a chance de se encontrar uma solução para o problema inteiro.

O Grupo 3 apresenta a busca por profundidade utilizando cada um dos cortes separadamente. Isto pode ser visto na Tabela 4.

Na Tabela 4, assim como na Tabela 3, os maiores tempos de resolução pertencem ao Clique e ao Corte fracionário de Gomory. Embora a melhor solução para este grupo tenha sido encontrada pelo Corte Fracionário de Gomory, este levou um tempo superior a 18 horas. O mesmo valor da função objetivo foi encontrado utilizando os cortes de Cobertura de Fluxo, e estes levaram menos que 5 minutos. Alto tempo de resolução não significa, necessariamente, que a melhor solução será encontrada, como no caso do corte Clique que precisou de um tempo superior a 6 horas para encontrar a pior função objetivo do Grupo 3.

Com relação ao método de escolha de nós, pode-se observar que a escolha por profundidade deixa menos nós ativos quando comparado com a busca pelo melhor limitante, isto pode ser observado comparando o corte Clique entre as tabelas 3 e 4. A busca pelo melhor limitante fica com 8 vezes mais

Tabela 4: Resultados do Grupo 3 - busca por profundidade

Planos de corte	Tempo (s)	Solução encontrada	Número de soluções encontradas	Nós explorados	Nós restantes	Iterações	Número de desigualdades
Clique	22946	781	55	1442255	1048957	6275988	12
Cobertura	360,36	772	82	379015	45930	2727751	420
Cobertura de fluxo	258,05	759	35	282129	115308	1699710	162
MIR	122,86	776	36	101964	8684	804665	434
Metade zero	917,34	769	89	723430	194228	6417188	668
Fracionários de Gomory	66606	759	119	2746830	2127516	13287503	129

nós ativos inexplorados. Computacionalmente, isto significa um maior número de recursos exigido do computador para memorização, o que pode gerar ausência de memória computacional.

A Figura 1 exibe uma comparação entre o número de soluções encontradas entre os grupos 2 e 3.

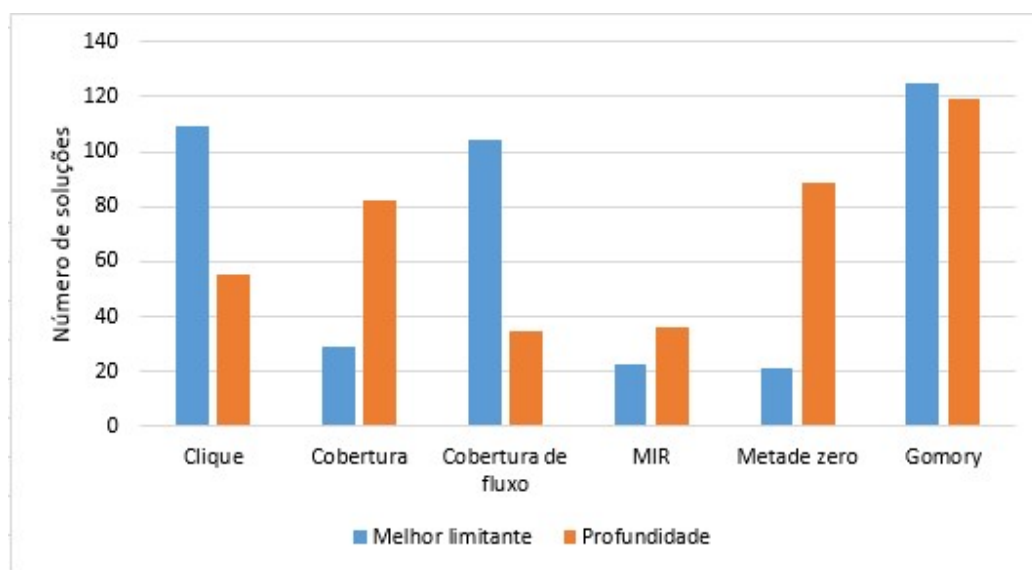


Figura 1: Número de soluções encontradas.

Ao comparar cada um dos 4 planos de corte com menor tempo de processamento (cobertura, Cobertura de Fluxo, MIR e Metade Zero) entre os Grupos 2 e 3, nota-se que na busca por profundidade apenas o Cobertura de Fluxo não apresenta o maior número de soluções. No entanto, os procedimentos mais longos (Clique e Corte Fracionário de Gomory) apresentam um maior número de soluções para busca pelo melhor limitante. Isto leva a concluir que quanto maior o tempo, maior a chance do método de busca em profundidade ficar em algum nó e não progredir. No entanto, em problemas rápidos, mais soluções são encontradas pela busca em profundidade, pois quanto mais baixo é o nível do nó na árvore de soluções, maior a chance de se encontrar uma solução inteira.

5. Conclusões

A aplicação de procedimentos para diminuição do tempo de resposta deve ser feita sempre que o tempo computacional for uma restrição. Em problemas clássicos, especialmente os de grande porte, é intrínseco a necessidade de métodos capazes de proporcionar boas soluções em tempos aceitáveis.



A escolha entre qual critério de nó a ser utilizado depende da dimensão do problema. Buscas em profundidade tendem a apresentar um maior número de soluções encontradas para problemas de rápida resolução quando comparadas com a busca pelo melhor limitante, isto devido a sua particularidade de explorar a fundo os nós, aumentando assim a chance de atingir uma solução inteira. No entanto, é possível que o programa fique em um nó por um longo período de tempo, fazendo com que em programas de grandes dimensões a busca pelo melhor limitante obtenha mais soluções, já que esta muda de região da árvore de soluções sempre que se tem um melhor limitante.

Utilizar *solvers* automaticamente, ou seja, sem alterar as configurações padrões, pode ser um bom procedimento quando não se tem conhecimento sobre as especificações do problema. No entanto, deixar o CPLEX resolver o problema nas configurações padrões não garante, por exemplo, a melhor escolha de composição dos planos de corte.

Para trabalhos futuros é sugerido o acréscimo de restrições ao problema, como distância entre os armazéns e o ponto de origem e a impossibilidade de alguns itens serem direcionados para alguns armazéns. Sugere-se também testar outras combinações de métodos, como diferentes critérios na escolha de nós e planos de corte adicionais.

Agradecimentos

Agradecemos a CAPES pelo auxílio financeiro.

Referências

- ARENALLES, M. et al. **Pesquisa Operacional**. Rio de Janeiro: Elsevier, 2006.
- BARRETO, A. M. F.; ARAÚJO, F.; PISCO, V. C.; Política de estoque e armazenagem para redistribuição de itens em uma loja. In: XXIII Simpósio de Engenharia de Produção, 23, 2016. Bauru **Anais...** Bauru: SIMPEP, 2016.
- EL AFIA, A.; KABBAJ, M. M. Supervised learning in Branch-and-cut strategies. In: Proceedings of the 2nd international Conference on Big Data, Cloud and Applications, 2., 2017, Tetouan, **Anais...** Tetouan: ACM, 2017. p. 114.
- HE, He.; DAUME III, H.; EISNER, J. M. Learning to search in branch and bound algorithms. In: Advances in neural information processing systems, 27., 2014, Montréal, **Anais...** Montréal: 2014, p. 3293-3301.
- HILLIER, F.; LIEBERMAN, G. **Introduction to Operations Research**. 9 ed. São Paulo: McGraw Hill. 2012. 1102 p.
- LAND, A. H.; DOIG, A. G. An automatic method of solving discrete programming problems. **Econometrica: Journal of the Econometric Society**, p. 497-520, 1960.
- QIN, Z.; HOU, Y.; LIU, S.; YAN, J.; LI, D. A branch-and-cut method for computing load restoration plan considering transmission network and discrete load increment. In: Power Systems Computation Conference (PSCC), 2014, Wroclaw. **Anais...** Wroclaw: IEEE, 2014. p. 1-7.
- SOUSA, A. S.; ASADA, E. N. Uma nova abordagem branch and cut aplicada ao problema de planejamento da expansão de redes de transmissão de grande porte. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, v. 23, n. 1, p. 108-119, 2012.